



Universidad de Costa Rica  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica



**IE-0117 Programación Bajo Plataformas Abiertas**

MSc. Marco Villalta Fallas - II Ciclo 2021

---

Laboratorio #6

Punteros, compiladores, herramientas y automatización de compilación.

---

**Instrucciones Generales:**

Este laboratorio se puede realizar de manera individual o en parejas.

El laboratorio debe entregarse antes del 16 de noviembre a las 23:59.

Utilice capturas de pantalla para demostrar la funcionalidad.

Entregue un archivo comprimido que incluya un directorio llamado **informe** con los archivos necesarios para generar el PDF del informe (.tex, imágenes, código, entre otros) y un directorio llamado **src** con los archivos de código fuente que lleven a la solución. Cualquier otro formato o entrega tardía no se revisará y el laboratorio tendrá una nota de cero.

Documente e investigue los algoritmos solicitados, en el reporte final se debe incluir un diagrama de flujo de la implementación realizada. Cualquier programa que no se compile correctamente no tendrá puntaje. En la calificación de este laboratorio se tomará en cuenta el uso del control de versiones git y la documentación en Doxygen

## 1. Compiladores (18 pts)

Conteste las siguientes preguntas de forma clara y concisa.

1. En su forma más general, en una sola palabra, ¿qué es un compilador?
2. Por medio de un diagrama, represente la principal diferencia entre un interprete y un compilador.
3. Explique brevemente con una o más razones porque un lenguaje compilado puede ser más eficiente que uno interpretado.
4. ¿A que se le conoce como cross-compiler?
5. Enumere los 5 elementos que componen la estructura típica de un compilador.
6. Describa brevemente la función que realiza, en el proceso de compilación, cada uno de los elementos mencionados en el punto anterior.

## 2. Depuración: gdb (28 pts)

1. ¿Cuál es la bandera del compilador gcc que se utiliza para generar un código que tenga los componentes necesarios para depurar (*debugging*)?
2. ¿Cuál es la línea de comandos en terminal necesaria para compilar con gcc un código fuente con el nombre **fuentes.c** en un ejecutable llamado **bin** capaz de ser depurado con gdb?
3. Describa brevemente para que sirven los siguientes comandos en gdb:

- run
- break
- quit
- continue
- print

4. Realice la depuración del programa `serie.c` proporcionado por el profesor a través de la herramienta `gdb`.  
**Nota:** Si el programa funcionara correctamente, debería de imprimir en consola el resultado de la serie  $(X^0)/0! + (X^1)/1! + (X^2)/2! + (X^3)/3! + (X^4)/4! + \dots + (X^n)/n!$  dados los valores  $X$  y  $n$ . El programa no da el resultado esperado y se sospecha de error(es) lógico(s). Muestre a través de capturas de pantalla el proceso realizado utilizando `gdb` para depurarlo.

### 3. Automatización de proyectos: make (10 pts)

1. Realice una estructura de directorios y archivos como la que se muestra a continuación:

```

.
|-- Makefile
|-- build
|-- include
|   |-- funciones.h
|-- src
|   |-- funciones.c
|   |-- main.c
3 directories, 5 files

```

2. Descomponga el programa `serie.c` en tres archivos. Uno debe de ser el `main.c` encargado de llamar las funciones, las cuales se encuentran definida en `funciones.h` e implementada en el archivo `funciones.c`, para calcular el el resultado de la serie.
3. Automatice el proceso de compilación realizando un Makefile para el ejercicio. Los archivos objeto (ejecutables) deben de quedar en el directorio `build`, para no popular los directorios de código fuente.
4. Comprima por aparte los ejercicios realizados de Makefile e incluyalos juntos con los archivos `.tex` utilizados para generar su informe final del laboratorio

## 4. Punteros

### 4.1. Punteros simples (9 pts)

Escriba un programa que cumpla con lo siguiente:

- Debe contener una función que acepte dos variables (por valor) del mismo tipo de dato (enteros o doubles) y trate de intercambiar sus valores. La función no puede imprimir las variables y tiene que devolver nada.
- Debe contener una función que acepte dos variables (por punteros) del mismo tipo de dato (enteros o doubles) y trate de intercambiar sus valores. La función no puede imprimir las variables y tiene que devolver nada.
- El programa debe crear 2 variables para cada funcion y debe imprimir sus contenidos antes y despues de llamar cada funcion. Explique lo que esta sucediendo.

### 4.2. Punteros, funciones y matrices. (15 pts)

Realice un programa que cumpla con:

- Inicialice e imprima los elementos de una matriz 10x10 de forma aleatoria con numeros enteros.
- Debe tener una funcion llamada `buscar` que encuentre la primer coincidencia de un numero solicitado al usuario en la matriz.

- La funcion *buscar* debe devolver la direccion de la primer coincidencia.
- No se puede utilizar los parentesis cuadrados [] dentro de la funcion. *buscar*.
- El prototipo de la funcion es:

```
int* buscar(int matrix[][COLS], int filas, int columnas)
```

## 5. Punteros de funciones (20 pts)

Realice un programa que calcule la integracion numerica de funciones matematicas utilizando el metodo de la regla del trapecio por lo que se debe cumplir con los siguientes requisitos:

1. Realice una funcion de C que calcule la funcion lineal  $y = x$ , que reciba como parametro un numero flotante(x) y devuelva otro numero flotante(y). El prototipo de la funcion es:

```
float flineal(float x);
```

2. Realice una funcion de C que calcule la funcion cuadratica  $y = x^2$ , que reciba como parametro un numero flotante(x) y devuelva otro numero flotante(y). El prototipo de la funcion es:

```
float fcuadratica(float x);
```

3. Realice una funcion de C que calcule la funcion polinomial  $y = x^3 + x^2 + x$ , que reciba como parametro un numero flotante(x) y devuelva otro numero flotante(y). El prototipo de la funcion es:

```
float fpolinomial(float x);
```

4. Realice una funcion de C que calcule la integral numerica con el metodo del trapecio desde un limite inferior  $a$  hasta un limite superior  $b$  de una funcion matematica pasada como un puntero de funcion. El prototipo de la funcion es:

```
float fintegraltrapecio(float a, float b, float (*f)(float));
```

5. Debe solicitar al usuario el limite inferior  $a$ , el limite superior  $b$  e imprimir el resultado de la integral para la funcion lineal, cuadratica y polinomial para los limites dados.